

**OBJECT ORIENTED APPARATUS AND METHOD FOR ALLOCATING
OBJECTS ON AN INVOCATION STACK IN A DYNAMIC
COMPILATION ENVIRONMENT**

ABSTRACT OF THE DISCLOSURE

5 An object oriented mechanism and method allow allocating Java objects on a
method's invocation stack in a dynamic compilation environment under certain
conditions. When a class is dynamically compiled by a just-in-time (JIT) compiler (as the
program runs), one or more of its methods may create objects that may be placed on the
method's invocation stack. During the compilation of the class, only the information
10 relating to the previously-loaded classes is taken into account. After compilation, as each
new class is loaded, the class is analyzed to see if loading the class might change the
analysis used to allocate objects on the invocation stacks of previously-compiled
methods. If so, the previous object allocations are analyzed in light of the object
reference(s) in the newly loaded class, and the previous object allocations are changed
15 from the invocation stack to the heap, if required. In this manner objects may be
allocated to a method's invocation stack based on information that is available from the
classes that have been loaded, and can then be changed to be allocated from the heap if
information in new classes shows that the previous decision (to allocate on the invocation
stack) is no longer valid.